

Paper 180-2008

Connecting to the Mainframe with SAS[®] Enterprise Guide[®]

Cindy Bailey Matthews, SAS Institute Inc., Cary, NC

ABSTRACT

SAS Enterprise Guide provides a point-and-click, intuitive interface to analyze your data no matter what system that data is stored on. But there are some system dependencies that you cannot completely ignore. In particular, z/OS (also known as "the mainframe") has some unique characteristics that can trip up the unwary SAS Enterprise Guide user. This paper will discuss the challenges of z/OS file systems, memory management, communication methods, and database access when used in conjunction with SAS Enterprise Guide. The topics were chosen from the most commonly asked questions to SAS Technical Support; for example:

- How do I increase the size of my WORK library space?
- How do I save my project in a z/OS PDS?
- How do I get more memory for the workspace server?
- How do I define a SAS Workspace Server to Workload Manager?

INTRODUCTION

SAS Enterprise Guide is part of a client/server environment. This paper focuses specifically on SAS Enterprise Guide and a SAS server in the mainframe (z/OS) environment. Some unique characteristics of this type of environment are controlled on the mainframe (z/OS) server side, and some are controlled by the client (SAS Enterprise Guide). With that in mind, this paper, discusses some of the challenges of using z/OS file systems, memory management, communication methods, and so on that can cause problems for unwary SAS Enterprise Guide users. This paper addresses common issues, or system dependencies, according to which machine (client or server) controls them.

Some of the details for each issue or dependency that is discussed vary according to the type of repository SAS Enterprise Guide uses. Unless otherwise stated, this paper assumes that the following is true:

- SAS Enterprise Guide is using a metadata repository.
- Your machine has SAS 9.1.3, Service Pack 4.
- Your copy of SAS Enterprise Guide 4.1 has the latest hot fix, 41EG06, applied.

CLIENT-SIDE ISSUES

Not all client-side issues that are controlled by the client machine are unique to SAS servers in a mainframe (z/OS) environment. However, the way they are handled in that environment can be unique. This section explains the EGAuto.sas built-in SAS[®] System program file, and it also discusses the following client-side tasks:

- defining bound libraries
- storing results
- displaying results and server memory
- saving or opening SAS Enterprise Guide projects or code
- handling DBMS libraries (DB2)
- creating stored processes

EXPLAINING EGAUTO.SAS

EGAuto.sas is a built-in SAS System program file that is executed when you use SAS Enterprise Guide to make a connection to the server. In releases of SAS Enterprise Guide prior to release 4.1, the system does not execute EGAuto.sas until you first execute code on the SAS System server.

Following is the path for the EGAuto.sas file SAS Enterprise Guide 4.1 is on the client machine:

```
C:\Documents and Settings\YOUR_USER_ID\Application Data\SAS\Enterprise Guide\4
```

Various macros are defined in EGAuto.sas. Three of these macros are discussed in this section. Two of the macros are specifically for the SAS server under z/OS. The other macro has unique logic for SAS servers under z/OS.

%ECLIBASSIGN and %ECLIBUNASSIGN Macros

The %ECLIBASSIGN and %ECLIBUNASSIGN macros are specifically for SAS servers under z/OS. These macros are intended to be used to establish or clear librefs on a SAS server under z/OS. %ECLIBASSIGN and %ECLIBUNASSIGN check the state of the libref before they actually define or clear the library (using the LIBNAME function).

%ENTERPRISEGUIDE Macro

Although it is not specifically for z/OS, the %ENTERPRISEGUIDE macro has unique logic for SAS servers under z/OS. This macro defines the variable SASWORKLOCATION, which contains the file path of a temporary z/OS data set. This macro variable serves as a place to which to direct graphics output (GIF files) when SAS Enterprise Guide downloads them, as needed, to the client machine. The file reference for SASWORKLOCATION is EGTMPDIR. If the SAS option FILESYSTEM= is set to MVS (the default value), you might encounter SVC99 allocation errors similar to the following:

```
ERROR: SVC99 error rc=4, reason=1708 : IKJ56228I DATA SET
userid.EGTEMP.SYS07067.T164558.PDSE NOT IN CATALOG OR CATALOG CAN NOT
BE ACCESSED.
```

This failure can occur if your site has restrictions on how z/OS temporary data sets can be allocated; for example, temporary data sets can only be allocated on certain direct access storage device (DASD) volumes. The location of this temporary file is based on the FILEDEV= and the FILEVOL= SAS options. The workaround for this problem varies, depending on your site's restriction. For example, the workaround might be as simple as adding the VOLUME= option in the FILENAME statement in EGAuto.sas, as shown in this example:

```
%global sasworklocation;
%if &sysscp=OS %then %do; /* z/OS Server */
  %if %sysfunc(getoption(filesystem))=MVS %then %do;
    /* By default, the physical file name will be considered a classic z/OS
       data set. */

    /* Construct a data set name that will be unique for each concurrent
       session under a particular account: */

    filename egtemp '&egtemp' disp=(new,delete); /* Create a temporary data
                                                set. */
    %let tempdsn=%sysfunc(pathname(egtemp)); /* Get the data set name. */

    filename egtemp clear; /* Remove the data set. We only wanted its
                           name. */
    %let unique_dsn="&EGTEMP.&substr(&tempdsn, 1, 16).PDSE";

    filename egtmpdir &unique_dsn
                     disp=(new,delete,delete) space=(cyl,(5,5,50))
                     dsorg=po dsntype=library recfm=vb
                     lrecl=8000 blksize=8004;
                     options fileext=ignore;
  %end;
  %else %do;

    /* By default, the physical file name will be
       considered a hierarchical file system (HFS)
       file. Note: This does NOT support users who
       do not have an HFS home directory. It also
       might not support multiple simultaneous
       sessions under the same account. */

    filename egtmpdir './';
  %end;
  %let path=%sysfunc(pathname(egtmpdir));
  %let sasworklocation=%sysfunc(quote(&path));
%end; /* MVS Server */
```

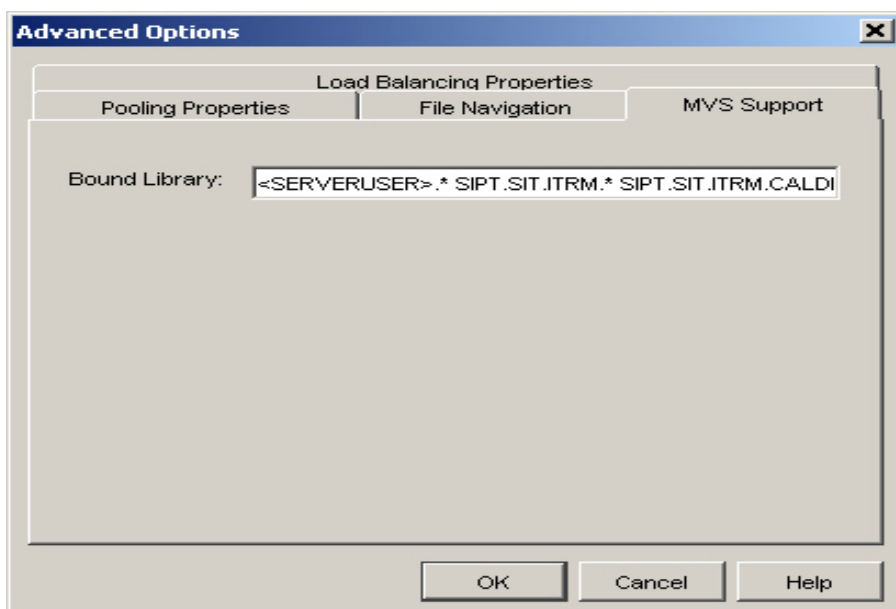
DEFINING BOUND LIBRARIES

When you view the **Server List** from SAS Enterprise Guide, you will see three types of files listed for each SAS Workspace Server under z/OS:

- libraries
- bound libraries
- files

A *bound library* is a list of SAS libraries that begin with a specific prefix or high-level qualifier. Bound libraries are reserved for SAS servers under z/OS only.

When you define a bound library on the **MVS Support** tab for the server metadata (either in SAS[®] Management Console or in the SAS Enterprise Guide Explorer), you can list multiple high-level qualifiers, up to a total of 256 characters.



Display 1. Defining a Bound Library

Each high-level qualifier is separated by a blank space, as shown in this example:

```
<SERVERUSER>. * sipt.itrm. * sipt.itrm.test. * dipt.sit.itrm. * sipt.sit.itrm.webpdb. *
```

Note: <SERVERUSER> is a wildcard that substitutes the server logon user ID. This substitution gives individual users more flexibility to see their own bound libraries in addition to the high-level qualifiers in the static list. You can find details on other wildcards in *Administering SAS Enterprise Guide 4.1* (support.sas.com/documentation/onlinedoc/guide/admin4.pdf).

STORING RESULTS

By default, SAS Enterprise Guide stores output data in one of three libraries on the SAS server: SASUSER, WORK, or EGTASK. (For more information about these libraries, refer to the section “[Library and File Allocation](#)”, later in this paper.) By default, if results are stored in either the SASUSER or the WORK library, the results will be lost at the end of the session because these SAS libraries are temporary z/OS data sets.

The EGTASK library, however, can be set up as a permanent library. So if you want a permanent location for your results, you can save them to EGTASK. You can change the default library for storing results by selecting **Tools ► Options ► Output Library** in SAS Enterprise Guide.

For more details on libraries, refer to *Administering SAS Enterprise Guide 4.1* (support.sas.com/documentation/onlinedoc/guide/admin4.pdf).

DISPLAYING RESULTS AND SERVER MEMORY USAGE

From SAS Enterprise Guide, you can choose whether to display your results back to SAS Enterprise Guide. You can also select the format that is used to display your results by selecting **Tools ► Options ► Results General**. However, these functions do use memory and storage on the server to buffer the results that are being displayed. Results that are displayed in text format use the least memory. If you suspect that you have memory-constraint problems, you can temporarily disable or alter these settings. To avoid directing the results back to SAS Enterprise Guide:

1. Select **Tools ► Options ► SAS Programs**
2. Under the **General** category in the right pane, deselect the check box **Automatically direct results back to SAS Enterprise Guide**.

SAVING AND OPENING SAS ENTERPRISE GUIDE PROJECTS OR CODE

You can save SAS Enterprise Guide projects and code to (or open existing projects and code from) a z/OS sequential file, a partitioned data set (PDS), or a partitioned data set extended (PDSE). To perform these tasks, you need to define the file navigation path in the metadata for the server. For SAS servers under z/OS, this path is a single path that consists of one or more high-level qualifiers. This path is the only one that you can save your projects or code to or from which you can open existing projects or code.

To open a project from a PDS, a PDSE, or sequential file:

1. Select **File ► Open ► Project ► Select SAS Servers/Folders ► Servers**.
2. Drill down to the appropriate server and file location.

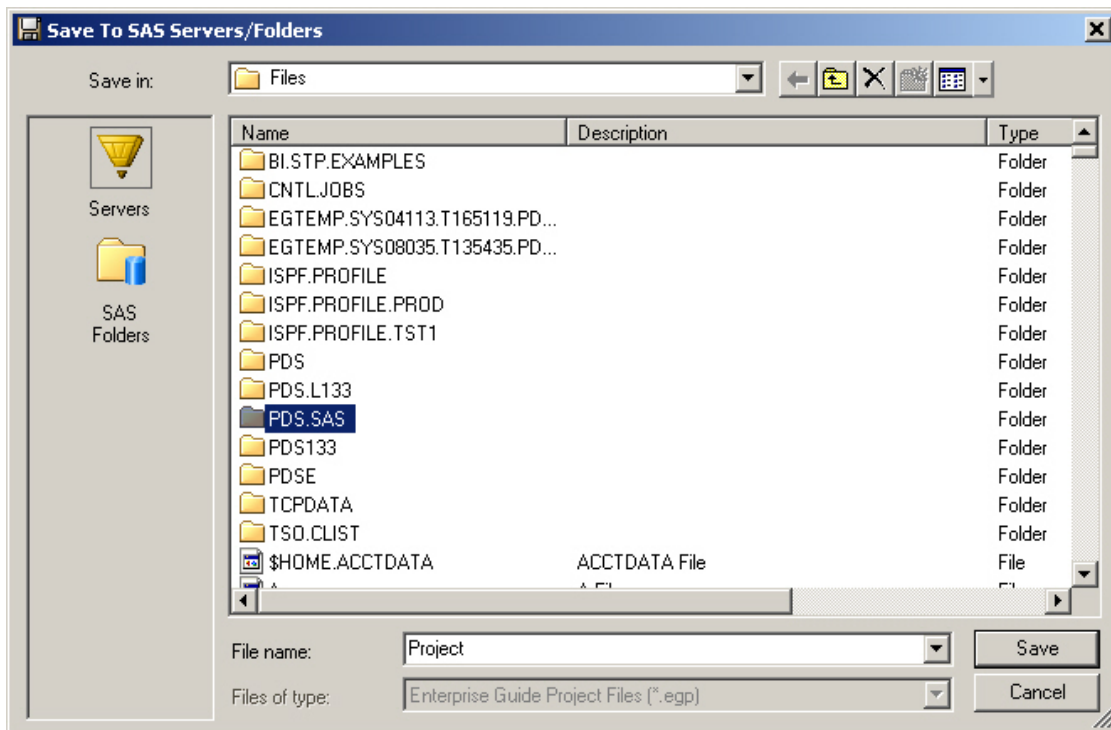
To save a project to a PDS, a PDSE, or sequential file:

1. Select **File ► Save Project As ► Select SAS Servers/Folders ► Servers**.
2. Drill down to the appropriate server and file location.

If the SAS option NOS99NOG is in effect, the server will recall any migrated files. If the SAS option S99NOMIG is in effect, the following error will occur:

```
ERROR: The file associated with this element could not be found. The file xxxxxxxx
could not be cached for Local use.
```

Normally, you save SAS Enterprise Guide projects to an existing PDS, PDSE, or sequential file. SAS Enterprise Guide can also dynamically allocate a sequential file to save a project to, but you can specify only one additional high-level qualifier in addition to the file navigation path high-level qualifier or qualifiers.



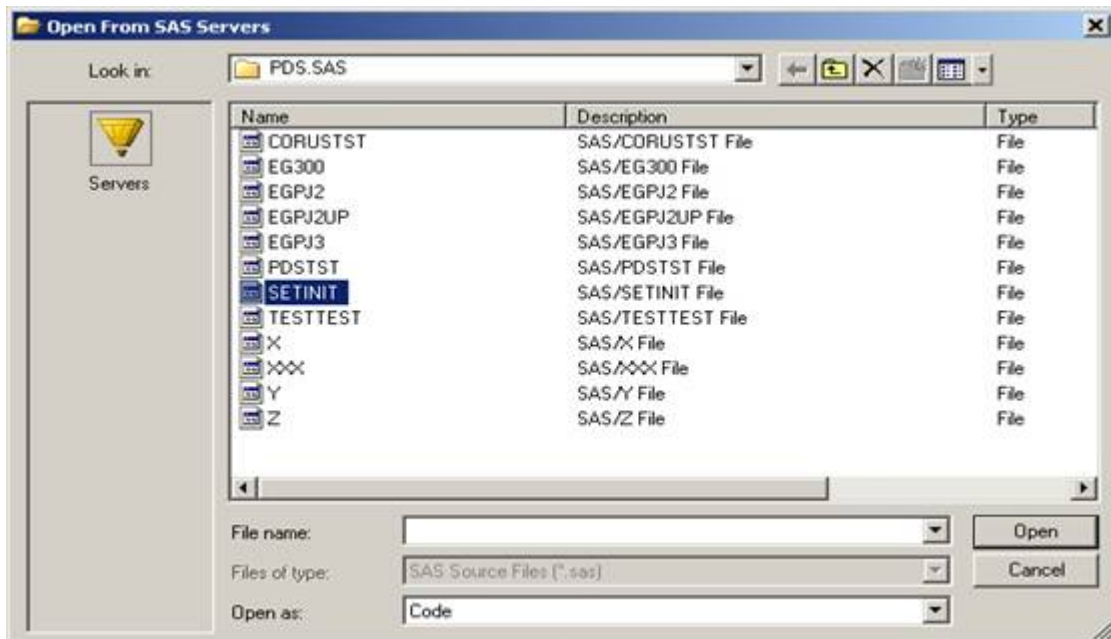
Display 2. Saving a SAS Enterprise Guide Project

To save code to a PDS, a PDSE, or a sequential file:

1. Select **File ► Save Code As ► Select SAS Servers/Folders ► Servers**.
2. Drill down to the appropriate server and file location.

To open code from a PDS, a PDSE, or a sequential file:

1. Select **File ► Open ► Code ► Select SAS Servers/Folders ► Servers**.
2. Drill down to the appropriate server and file location.



Display 3. Opening SAS Enterprise Guide Code

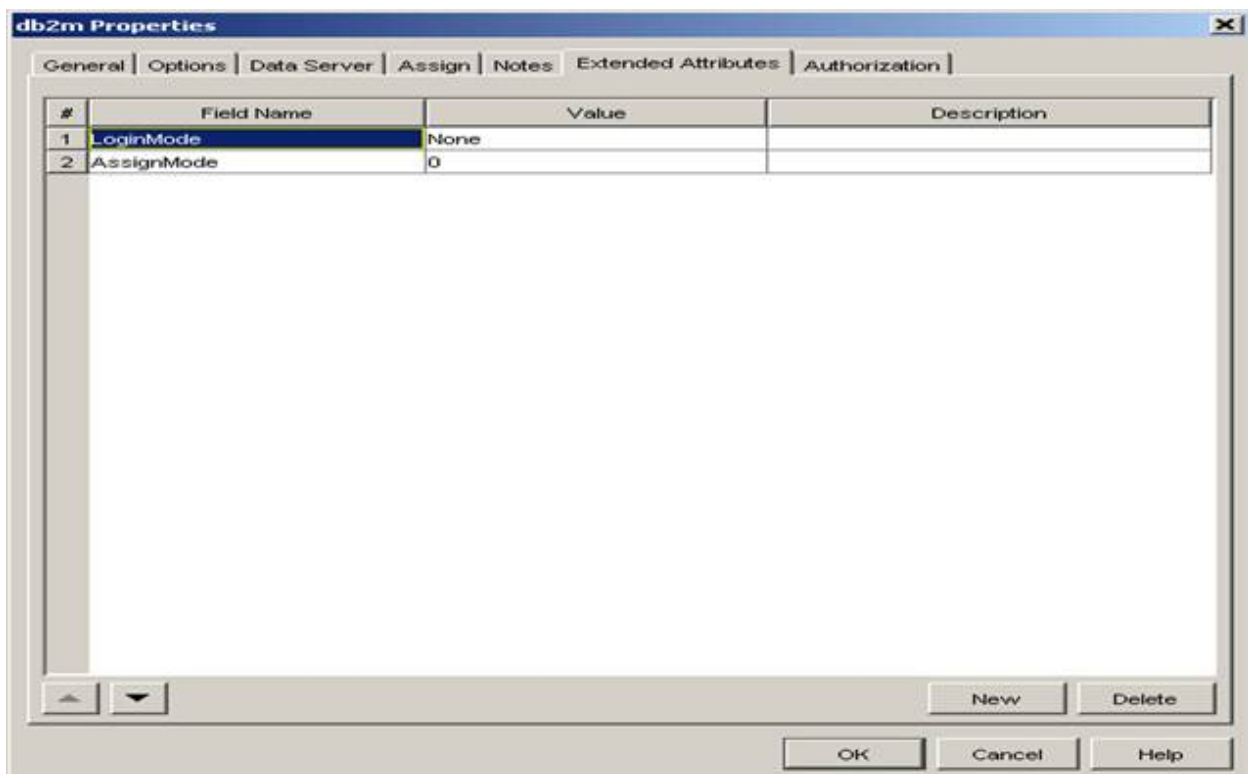
HANDLING DBMS LIBRARIES (DB2)

DB2 tables are handled differently under z/OS than other DBMS tables handled on other platforms. When you define DB2 tables that will be accessed by SAS Enterprise Guide, you must specify two metadata library properties, **LoginMode** and **AssignMode**. The values for **LoginMode** and **AssignMode** should be **None** and **0**, respectively.

For details on how to define these two metadata library properties, refer to the following documents:

- *What SAS® Administrators Should Know about Libraries, Metadata, and SAS® Enterprise Guide® 4.1* (support.sas.com/techsup/technote/ts788.pdf).
- Usage Note 31095 (support.sas.com/kb/31/095.html)

After you have assigned the values for these metadata properties, those assignments appear on the **Extended Attributes** tab in the library's Properties dialog box, which you can view from SAS Management Console, as shown in the following example. You can view or modify these properties from the **Extended Attributes** tab.

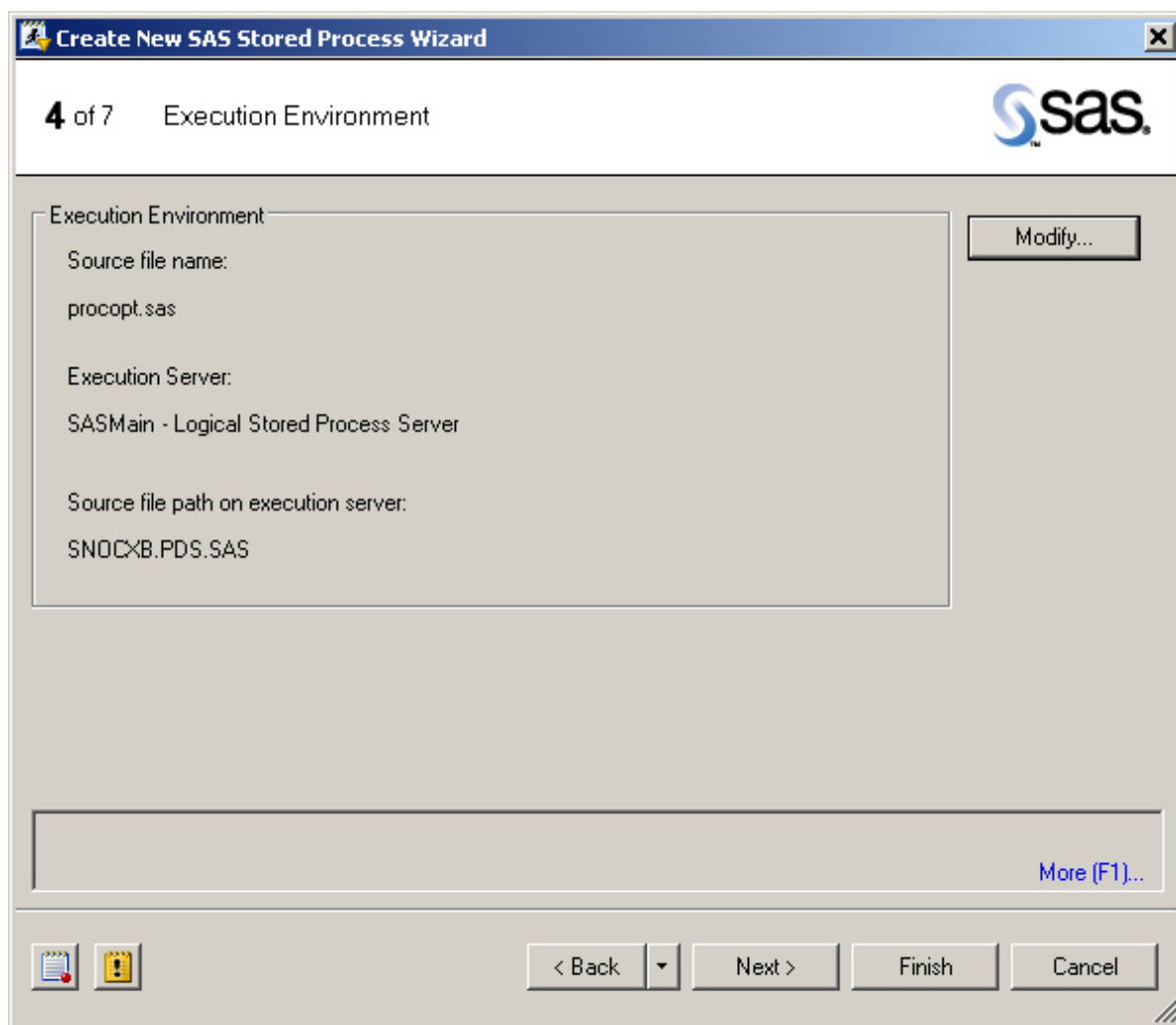


Display 4. Assigning Metadata Library Properties

CREATING STORED PROCESSES

When you are using a SAS server under z/OS, stored processes are created the same way as on other hosts. However, you can have a PDS, a PDSE, an HFS, or a zFS file as the source code repository.

When you create a stored process in the Create New SAS Stored Process Wizard (step 4 of 7), the source filename is listed with a .sas extension, as shown in the following example.



Display 5. Source Filename (procopt .sas) with the .SAS File Extension

If you save the stored process to a PDS or PDSE, this extension is ignored, and it is not included in the PDS or PDSE member name. If you save the source to an HFS or a zFS file, the extension is included with the stored process name.

SERVER-SIDE ISSUES

Although not all server-side issues are unique to mainframe servers, the ways in which they are handled on mainframe servers are unique. Server-side issues normally require the involvement of a z/OS systems programmer. This section discusses the following server-side issues:

- object spawner options
- memory allocations for spawned servers
- libraries and files allocation
- communication access methods
- Workload Manager
- timeouts for spawned servers
- SAS log for the Workspace Server
- UNIX System Services (USS) environment variables that might not be configured

OBJECT SPAWNER OPTIONS

Some object spawner options are not set by default, and they have unique functionality that SAS Technical Support often recommends to users for unique needs. This section discusses the most common of those options. If your site has a customized (nonstandard IBM) TCP/IP configuration, your TCP/IP environment might require the need to use one of these object spawner options. For example, if you receive an error that indicates a host-name resolution

failure, you might need to specify an additional object spawner option in order to correct the error. For details on the cause of this error, you can use a TCPTTEST procedure similar to the following:

```
filename sasctcpe 'userid.sasctcpe' disp=(new,catlg) recfm=fba
        lrecl=133; /* Update the data set name. */

proc tcptest;
run;
```

PROC TCPTTEST displays the results of the GETHOSTNAME and GETHOSTBYNAME TCP function calls, and it provides more details on the host-name resolution failure. Note: GETHOSTNAME retrieves a string that contains its host name from the TCP/IP stack. GETHOSTBYNAME resolves the host-name string to its IP address.

In this type of situation, you might need one of the following options. With a host-name resolution failure, it is not always clear and definitive which of these options you might need to use. Therefore, you should contact SAS Technical Support to help you evaluate the problem and determine the appropriate option to use.

-dnsMatch

specifies a Domain Name Service (DNS) alias that the object spawner will accept as a match for the local machine name. In addition, the spawner replaces the `-dnsMatch` value with the local machine name in its list of servers. This option is necessary if your network configuration resolves a single DNS alias to multiple machines that run SAS Object Spawners. For example, suppose you configure SAS servers and spawners on two different machines (*machinename1.my.org* and *machine-name2.my.org*). The DNS alias *srv.my.org* resolves to both of these machines. As a result, client machines can send a request to the DNS alias, and one of the two spawners will receive the request. To support this configuration, you need to specify the following option in the spawner start-up command on each machine.

```
-dnsMatch srv.my.org
```

-dnsName

specifies which IP stack is used for communication between the spawner and the servers that the spawner launches. This option can be abbreviated as `-dns`.

-hostKnownBy (load-balancing spawners only)

specifies a DNS alias that the object spawner will accept as a match for the local machine name. You might need to use this option along with the `lbUseHostName` option if your server machine is known to your client machines by a different machine name and IP address.

-lbUseHostName (load-balancing spawners only)

instructs the spawner to use machine names rather than IP addresses when it directs clients to SAS servers. If the IP address for your server machine is different for connections within your network than for connections outside of your network, then this option is necessary to enable connections from outside of your network.

Note: If you use this option, the machine names in your server definitions must resolve to the correct IP addresses at your client machines.

Another object spawner option that you might need is the `-allowxcmd` option. This option is not set by default and it enables the workspace server to issue host commands (such as TSO) and also PIPE commands for all servers that are started by the spawner. By default, the spawner starts all servers with the `-noxcmd` SAS system option. When you specify `-allowxcmd`, the spawner no longer specifies `-noxcmd` when it launches server sessions.

Caution: When you specify `-allowxcmd`, client machines can use host commands to perform potentially **harmful** operations such as file deletion.

An example of when you need this option is when you need to allow the workspace server to call external programs from SAS code. If you attempt to run code that requires this option and the option is not set, you will receive the following error:

```
ERROR: System command execution is not authorized.
```

The following example illustrates code in which you would supply one of the previously discussed options. In this example, *object-spawner-option* is a user-supplied value for which you would substitute the appropriate option that your situation requires.


```
//OBJSPAWN PROC PROG=OBJSPAWN,
//      OPTIONS='-CONFIGFILE /usr/lpp/SAS/SAS_9.1/objspawn913.conf ',
//      OPT2='-SV -SLF LOGFILE object-spawner-option'
```

For a complete list of object spawner options, refer to the SAS® 9.1.3 *Integration Technologies Server Administrator's Guide, Third Edition* (support.sas.com/rnd/itech/updates/913/admin_oma_sp4.pdf).

MEMORY ALLOCATION FOR SPAWNED SERVERS

The spawned SAS Workspace Server is a UNIX System Services (USS) process, and it follows all USS rules that govern memory/storage allocation for a USS process. The order of precedence (lowest to highest) for allocating memory for a USS address space is:

- the MAXASSIZE parameter in the 'SYS1.PARMLIB(BPXPRMxx)' file-
- the ASSIZEMAX parameter specified in the RACF USS/OMVS segment that is associated with the authenticated user
- exits (for example, IEFUSI) that affect USS processes-

LIBRARY AND FILE ALLOCATION

In a client/server environment, you need to be aware of some of the unique characteristics and processes that are related to allocating libraries and files.

In SAS Enterprise Guide, you store results from queries or tasks in one of following three SAS libraries, depending on the order of precedence set in the software.

- **SASUSER library**—Unlike the SASUSER library in a TSO SAS session, a SAS Workspace Server's SASUSER library is not allocated, by default, in a z/OS Workspace Server environment. The SASUSER library is not allocated in order to avoid contention with users logging on directly to z/OS and running an interactive SAS session under TSO. All references to SASUSER are directed to the WORK library, which is a temporary z/OS file, so the data will not be permanently stored or saved.
- **WORK library**—For a SAS server in the z/OS environment, the WORK library is allocated differently than in a batch or TSO SAS session. By default, this library is allocated based on the values for the SAS options FILESPPRI, FILESPSEC, FILEDEV, and FILEUNIT. See SAS Usage Note 9531 (support.sas.com/kb/9/531.html) for details on setting these options.

The WORK library can also be directed to an HFS/zFS directory. For details on this setup, see SAS Usage Note 11159 (support.sas.com/kb/11/159.html).

In addition, SAS 9.1 and earlier versions do not allow the WORK library in a server environment to be in a multivolume SAS library. If you need to use a multivolume SAS library, you can either use a SAS library other than WORK or you can allocate a different z/OS temporary library and use the SAS USER= option in the OPTIONS statement to redirect the WORK Library, as shown in this example:

```
libname x '&cindy';
libname x list;      /* The physical file is a z/OS temporary data set. */
options user=x;

data testcode;
  do i=1 to 10;
    output;
  end;
run;

proc contents data=x.testcode;
run;
proc contents data=testcode; /* Shows the same physical file as x.testcode.*/
run;
proc contents data=work.testcode; /* Receives "ERROR: File
                                   WORK.TESTCODE.DATA does not exist.",
                                   which indicates that the WORK library
                                   is not used.                               */
run;
```

- **EGTASK library**—The EGTASK library is not allocated, by default, in a SAS Workspace Server environment. If the SAS Enterprise Guide output results need to be stored in a permanent SAS library, you can allocate this EGTASK library in one of the following three ways:
 - Define the EGTASK library in the metadata. All connecting users will then access the same EGTASK library.
 - Dynamically allocate a library for each user by adding logic to the SAS command list (CLIST) or to the REXX exec that invokes the SAS Workspace Server. This additional EGTASK logic can be similar to the logic that is used in the SAS REXX exec and CLIST to allocate the SASUSER library.
 - Execute a LIBNAME statement for EGTASK to the server before running any code for the current SAS Enterprise Guide session.

You should also be aware of other library and file allocation issues that might occur, such as the following:

- **SAS library tapes**—SAS strongly recommends that you not run SAS code against SAS tape data sets from SAS Enterprise Guide because of the delay that occurs with mounting tapes and the fact that the tape drive is not released until the library reference is cleared or deallocated.

If you MUST use tape libraries or files with SAS Enterprise Guide, then your environment will have some special requirements. First, the authenticated user ID must have TSO mount authority. In addition, you must change the USS shell script that is used in the object spawner environment to a different USS utility (TSOCMD). You need to download this new utility from an IBM Web site and follow the setup instructions found in Usage Note 13612 (support.sas.com/kb/13/612.html).

Note: You will not see SAS tape libraries listed under the **Server List for Bound Libraries**. A *bound library* is a list of SAS libraries that begin with a specific prefix or high-level qualifier.

- **Migrated data sets**—The SAS system option S99NOMIG controls the behavior of migrated data sets under z/OS. S99NOMIG is a toggle option (S99NOMIG | NOS99NOMIG). The default is NOS99NOMIG, which enables the software to recall migrated data sets. If NOS99NOMIG is set for the server environment, the SAS Workspace Server will recall migrated data sets (SAS libraries and sequential files) that are defined in the repository and that are dynamically allocated with SAS code.

Note: Migrated SAS libraries are not listed under the Server List for **Bound Libraries**.

COMMUNICATION ACCESS METHODS

The default communication access method under z/OS is Cross Memory Services (XMS). However, the connection from SAS Enterprise Guide to the SAS Object Spawner under z/OS uses TCP/IP as the communication access method. From the z/OS workspace server (SAS® Stored Process Server), you can connect to other SAS servers (such as the SAS/SHARE® software server-) using either XMS or TCP/IP as the communication access method. You can use either of these communication access methods because support has been added to SAS 9.1.3, Service Pack 4 with Hot Fix E9BA73 so XMS can handle two-level server names, primarily for a Systems Complex (Sysplex) environment. For details on using the two-level server name, see Usage Note 18992 (support.sas.com/kb/18/992.html).

Note: Hot Fix E9BA73 is not available from the SAS Customer Support Web site. You must contact SAS Technical Support for the hot fix. If you are in North America, you can reach Technical Support at 919-677-8008. If you are outside of North America, you must contact your local SAS office.

WORKLOAD MANAGER

Users frequently ask SAS Technical Support how to define a SAS server to Workload Manager (WLM), but there is no definitive answer to that question. It is relative to the importance of the specific task and should be compared to the importance of other workloads on the system and what is needed to balance the overall capacity on the machine. The spawned SAS Workspace Servers (USS processes) are under control of the OMVS workload. To control the priority of the spawned workspace server, you should first establish the various WLM service classes (for example, OMVSLO, OMVSMED, and OMVSHI). Then, in WLM, you create an AI Classification Rule that matches the accounting information (that is, the `_BPX_ACCT_DATA` environment variable), which puts the task into the particular server class. You can establish the service classes in the object spawner's USS shell script by setting `_BPX_ACCT_DATA` to one of the WLM service classes, as shown in this example:

```
#
# Use the account data to place SAS in the correct WLM service class
#
export _BPX_ACCT_DATA=OMVSMED
```

Note: This topic also applies to the Intelligence Platform Servers (Workspace, Stored Process, and Connect).

You can find this information on controlling priority of spawned servers in Workload Manager in Usage Note 13230 (support.sas.com/kb/13/230.html).

TIMEOUTS FOR SPAWNED SERVERS

SAS[®]9 spawned servers will time out according to the SMFPRMxx JWT value in SYS1.PARMLIB. For example, if the JWT value is set to 20 minutes, a workspace server will abnormally end (with an S522 abend code) if the workspace server is left idle for more than 20 minutes. Note that this is true for the SAS Stored Process Server as well. However, the SAS Stored Process Server wakes up, by default, every 15 minutes for clean up. So, unless the JWT value is set to 15 minutes or less, the stored process servers should not time out as a result of JWT.

You can avoid such timeouts in one of three ways:

- Disable JWT completely for all USS processes by setting the MAXCPUPTIME parameter to 86400 in SYS1.PARMLIB. (This is much the same as setting TIME=1440 in JCL.)
- Disable JWT completely for a particular user by setting the OMVS CPUTIMEMAX value to 86400 in the user's RACF. (Again, much the same as setting TIME=1440 in JCL.) You set the value in the RACF as follows:

```
altuser userid omvs(cputimemax(86400))
```

- Use the IEFUTL exit to extend the wait time, which is a more flexible way to avoid a timeout. If IEFUTL needs to uniquely identify these SAS processes, then you have at least two choices for setting the account information.
 - You can set the accounting information in the shell script that starts the server using the `_BPX_ACCT_DATA` environment variable:

```
export _bpx_acct_data=sasws1
```

This setting makes the accounting information available to the IEFUTL exit.

- You can set the accounting information on an individual basis, as shown in the following RACF command:

```
altuser userid workattr(waacnt(sasws1))
```

You can also tailor the IEFUTL exit so that time is extended differently based on the time of day. For example, you might extend the time more during non-primetime hours.

You can find information about controlling timeouts in Usage Note 14666 (support.sas.com/kb/14/666.html).

SAS LOG FOR THE SAS WORKSPACE SERVER

The ONLY time you should obtain the SAS log for the SAS Workspace Server is when you are debugging a specific problem because log files grow quickly and can affect performance. For an environment that uses a metadata repository, you can obtain the SAS log for a workspace server by following these steps:

1. Open SAS Management Console.
2. In the tree view, select **Server Manager ► SASMain ► SASMain - Logical Workspace Server ► SASMain - Workspace Server**.
3. Right-click **SASMain - Workspace Server** and select **Properties**.
4. Select the **Options** tab and add the appropriate options to the launch command, which should look similar to the following:

```
/usr/lpp/SAS/BI/SAS_9.1/Levl/SASMain/startsas.sh
o('log="/tmp/objspawn_Wrkspcse_D%Y%m%d.log" verbose logparm="rollover=auto
open=replaceold write=immediate"') --
```

Note: Ensure that every user who might execute a workspace server session has Write permission to the directory where the logs will be written. Also note the use of double and single quotation marks in the command!

If SAS Enterprise Guide is using the EG repository (which is not part of an intelligence platform installation), you need to set the `sasCommand` attribute in the object spawner's configuration file. The `sasCommand` attribute should look similar to the following:

```
sasCommand: /usr/lpp/SAS/SAS_9.1/sasobj913.sh options('COMAMID=TCP
LOGPARM="ROLLOVER=SESSION WRITE=IMMEDIATE" ALTLOG="sequential.file" ') --
sasObjectServerParms: iomlevel=3 applevel=3
```

Notes:

- The entire `sasCommand` attribute must be on one line. The previous example does not appear on one line because of space limitations. However, the command **MUST** be on a single line in the CONFIG file.
- The sequential file must be a pre-allocated sequential file with `RECFM=FBA` and `LRECL=133`.
- The `LOG=` option will not work when you use a CLIST to execute the server.
- You must enclose the sequential file's filename in double quotation marks.

SPECIFYING UNIX SYSTEM SERVICES (USS) ENVIRONMENT VARIABLES THAT MIGHT NOT BE CONFIGURED

By default, one USS process is created by a connection between SAS Enterprise Guide and the SAS Object Spawner under z/OS. However, you might see additional processes based on statements in the USS shell script and the values that are specified for USS environment variables.

The reason more than one process is launched is due to the USS environment. If a shell script begins with `#!/bin/sh` (and it should ALWAYS appear at the beginning of a shell script), then the script will be interpreted as a shell program. However, when the USS spawner service spawns a shell script, that shell script does not run in a shell process. The shell runs `/etc/profile` and `~/profile` to customize the user environment. This customization does not occur, though, when a script is spawned, and some useful environment variables are not always configured. As a result, SAS recommends that you specify the following USS environment variables in the USS shell script for the object spawner:

```
export _BPX_SHAREAS=YES
export _BPX_BATCH_SPAWN=YES
export _BPX_SPAWN_SCRIPT=YES
```

Note that in 'SYS1.PARMLIB(BPXPRMxx)', the parameter `MAXPROCUSER` limits the number of processes per user ID. You need to consider this fact if your connecting user IDs are defaulting to the same RACF USS/OMVS segment that assigns the same USS user ID to each user ID. For more details, see Usage Note 16609 (support.sas.com/kb/16/609.html).

CONCLUSION

This paper covers specific issues involved in a server-client relationship between the z/OS operating environment (SAS Workspace Servers) and SAS Enterprise Guide. The intent of the paper is to reveal and clarify unique areas and issues related to using z/OS servers with SAS Enterprise Guide as the client.

ACKNOWLEDGMENTS

The author thanks the following co-workers in the SAS Technical Support Division for reviewing this paper and answering questions: Susan Marshall, Francisco Ojeda, Eli Kovick, and Jeff Stead. The author also appreciates the following people for their input and review: Cummins Mebane (SAS Enterprise Guide development), Keefe Hayes (mainframe development), Lindy Mayfield (SAS European Sales Support), and John Hamlet (z/OS systems programmer).

RECOMMENDED READING

SAS Institute Inc. 2008. *What SAS® Administrators Should Know about Libraries, Metadata, and SAS® Enterprise Guide® 4.1*. Available at support.sas.com/techsup/technote/ts788.pdf.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Cindy Matthews
SAS Institute Inc.
SAS Circle Dr.
Cary, NC 27513
Cindy.Matthews@sas.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. © indicates USA registration. Other brand and product names are trademarks of their respective companies.